IBM

APL2 Reference Card

APL2 Character Set

```
         0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
         0 1 2 3 4 5 6 7 8 9 A B C D E F
      +--------------------------------------+
00    |                                      |  00
10    |                                      |  10
20    |                                      |  20
30    |                                      |  30
40    |    A B C D E F G H I ¢ . < ( + |     |  40
50    |  & J K L M N O P Q R ! $ * ) ; ¬     |  50
60    |  - / S T U V W X Y Z | , % _ > ?     |  60
70    |    ` : # @ ' = "                     |  70
80    |  ~ a b c d e f g h i ↑ ↓ ≤ ⌈ ⌊ →    |  80
90    |  ⌷ j k l m n o p q r ⊃ ⊂ ○ ←        |  90
A0    |    ~ s t u v w x y z ∩ ∪ ⊥ ≥ ≡       |  A0
B0    |  α ∊ ⍳ ⍴ ⍵ × \ ↑ ∇ ∆ ⊤ ⌉ ≠ |        |  B0
C0    |  { A B C D E F G H I ⍟ ⍣ ⌹ ⌽ ⍉      |  C0
D0    |  } J K L M N O P Q R ! ♦ ⌽ ⍟ ⍝      |  D0
E0    |  \ ≡ S T U V W X Y Z / ⍫ ⍕ ⊖ ⍕ ⍒    |  E0
F0    |  0 1 2 3 4 5 6 7 8 9 ⍙ ⍋ ● ⍦        |  F0
      +--------------------------------------+
         0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
         0 1 2 3 4 5 6 7 8 9 A B C D E F
```

Primitive Functions

Symbol	Symbol Name	Monadic Function	Dyadic Function
+	Plus	Conjugate	Add*
-	Bar	Negative	Subtract*
×	Times	Direction	Multiply*
÷	Divide	Reciprocal	Divide*
		Stile	Magnitude	Residue*
⌈	Up Stile	Ceiling	Maximum*
⌊	Down Stile	Floor	Minimum*
?	Query	Roll	Deal*
*	Star	Exponential	Power*
⍟	Circle Star	Natural Log	Logarithm*
○	Circle	Pi Times	Circle Function*
!	Quote Dot	Factorial	Binomial*
⌹	Quad Divide	Matrix Inverse	Matrix Divide
~	Tilde	Not	Without
∧	Up Caret		And*

*Axis Permitted

2

Primitive Functions (continued)

Symbol	Symbol Name	Monadic Function	Dyadic Function
∨	Down Caret		Or*
⍲	Up Caret Tilde		Nand*
⍱	Down Caret Tilde		Nor*
<	Less		Less Than*
≤	Not Greater		Less Than or Equal*
=	Equal		Equal*
≥	Not Less		Greater Than or Equal*
>	Greater		Greater Than*
≠	Not Equal		Not Equal*
≡	Equal Underbar	Depth	Match
ρ	Rho	Shape	Reshape
⊃	Right Shoe	Disclose*	Pick
⊂	Left Shoe	Enclose*	
,	Comma	Ravel*	Catenate*
⌽	Circle Stile	Reverse*	Rotate*
⍉	Circle Slope	Transpose	Transpose
[ ]		Bracket Index	
↓	Down Arrow		Drop*
↑	Up Arrow	First	Take*
ι	Iota	Interval	Index of
⍷	Epsilon Underbar		Find
⍋	Delta Stile	Grade Up	Grade Up
⍒	Del Stile	Grade Down	Grade Down
∈	Epsilon	Enlist	Member
⍎	Down Tack Jot	Execute	
⍕	Up Tack Jot	Format	Format

*Axis Permitted

3

Primitive Functions (continued)

Symbol	Symbol Name	Monadic Function	Dyadic Function
⊥	Down tack		Decode
⊤	Up tack		Encode
⎕AF			Atomic Function
⎕AT			Attributes
⎕CR		Character Rep	
⎕DL		Delay	
⎕EA			Execute Alt
⎕EC		Execute Controlled	
⎕ES		Event Simulate	Event Simulate
⎕EX		Expunge	
⎕FX		Fix	Fix
⎕NA		Name Assoc	Name Assoc
⎕NC		Name Class	Name Class
⎕NL		Name List	Name List
⎕SVC		Shar Var Control	Shar Var Control
⎕SVO		Shar Var Offer	Shar Var Offer
⎕SVQ		Shar Var Query	
⎕SVR		Shar Var Retract	
⎕SVS		Shar Var State	
⎕TF			Transfer Form

*Axis Permitted

System Variables

⎕AI	Account Information	⎕PP	Print Precision
⎕AV	Atomic Vector	⎕PW	Print Width
⎕CT	Comparison Tolerance	⎕PR	Prompt Replacement
⎕EM	Event Message	⎕RL	Random Link
⎕ET	Event Type	⎕R	Right Argument
⎕FC	Format Control	⎕SVE	Shared Var Event
⎕IO	Index Origin	⎕TC	Terminal Control
⎕LX	Latent Expression	⎕TS	Time Stamp
⎕L	Left Argument	⎕TZ	Time Zone
⎕LC	Line Counter	⎕UL	User Load
⎕NLT	National Language	⎕WA	Workspace Available

4

Primitive Operators

Syntax	Name	Example
`LO/R` `LO#R`	Compress	`1 0 0 1 1/'PLEAT'` `PAT`

`LO/[X]R`
`LO#[X]R`
	Compress with Axis	`1 0 1/[1]3 2p16`
		`1 2`
		`5 6`

`L LO¨ R`	Each (Dyadic)	`1 2 3, ¨ 4 5 6` `1 4   2 5   3 6`
`LO¨ R`	Each (Monadic)	`p¨'TOM' 'DICK'` `3   4`

`LO\R`
`LO\R`
	Expand	`A ← 1 0 1 0 0 1`
		`A\1 2 3`
		`1 0 2 0 0 3`
		`A\'ABC'`
		`A B  C`

`LO\[X]R`
`LO\[X]R`
	Expand with Axis	`1 0 1\[1]2 3p16`
		`1 2 3`
		`0 0 0`
		`4 5 6`

`L LO.RO R`	Inner Product	`R ← 2 3 p 16`
		`L ← 2 2 p 10×14`
		`L +.×R`
		`90 120 150`
		`190 260 330`

`L ∘.RO R`	Outer Product	`(13) ∘.× 15`
		`1 2 3 4 5`
		`2 4 6 8 10`
		`3 6 9 12 15`

`LO/R` `LO#R`	Reduce	`+/1 2 3 4 5` `15`
`L LO/R` `L LO#R`	Reduce N-wise	`4+/16` `10 14 18`

`L LO/[X]R`
`L LO#[X]R`
	Reduce N-wise with Axis	`3+/[1]4 3 p 112`
		`12 15 18`
		`21 24 27`

`LO/[X]R` `LO#[X]R`	Reduce with Axis	`+/[1]3 4 p 112` `15 18 21 24`
`LO/R` `LO#R`	Replicate	`1 2 3 4/'ABCD'` `ABBCCCDDDD`

`LO/[X]R`
`LO#[X]R`
	Replicate with Axis	`T ← 2 3 p 16`
		`2 ¯1 0 1/[2]T`
		`1 1 0 3`
		`4 4 0 6`

`LO\R` `LO\R`	Scan	`+\1 2 3 4 5` `1 3 6 10 15`

`LO\[X]R`
`LO\[X]R`
	Scan with Axis	`+\[1]2 3 p 16`
		`1 2 3`
		`5 7 9`

Circle Functions

L	L○R	L	L○R
		0	$(1-R*2)*0.5$
¯1	Arcsin R	1	Sine R
¯2	Arccos R	2	Cosine R
¯3	Arctan R	3	Tangent R
¯4	$(¯1+R*2)*0.5$	4	$(1+R*2)*0.5$
¯5	Arcsinh R	5	Sinh R
¯6	Arccosh R	6	Cosh R
¯7	Arctanh R	7	Tanh R
¯8	$-(¯1-R*2)*0.5$	8	$(¯1-R*2)*0.5$
¯9	R	9	Real R
¯10	$+R$	10	$|R$
¯11	$0J1×R$	11	Imaginary R
¯12	$*0J1×R$	12	Phase R

Boolean Functions

In the following example, L and R have the following values:

Symbol Name	Function	Result			
		L←0 0 1 1			
		R←0 1 0 1			
	L	0 0 0 0			
And	$L∧R$	0 0 0 1			
Greater	$L>R$	0 0 1 0			
	L	0 0 1 1			
Less	$L<R$	0 1 0 0			
	R	0 1 0 1			
Not Equal	$L≠R$	0 1 1 0			
Or	$L∨R$	0 1 1 1			
Nor	$L⍱R$	1 0 0 0			
Equal	$L=R$	1 0 0 1			
	$~R$	1 0 1 0			
Not Less	$L≥R$	1 0 1 1			
	$~L$	1 1 0 0			
Not Greater	$L≤R$	1 1 0 1			
Nand	$L⍲R$	1 1 1 0			
	1	1 1 1 1			

Format (by Example) Control Characters

$L⍕R$
0-Pad zeros to this point
1-Float decorator if negative
2-Float decorator if nonnegative
3-Float decorator
4-Do not float nearest decorator
5-Normal digit
6-Right decorator ends field
7-Use next decorator to right for scaled form
8-Fill empty positions of field with ⎕FC[3]
9-Pad zeros to this position if nonzero
.-Decimal point ,-Controlled comma

Event Type Codes

□*ET* values	
0 0	No error
0 1	Unclassified event
1 1	*INTERRUPT*
1 2	*SYSTEM ERROR*
1 3	*WS FULL*
1 4	*SYSTEM LIMIT*-symbol table
1 5	*SYSTEM LIMIT*-no shares
1 6	*SYSTEM LIMIT*-interface quota
1 7	*SYSTEM LIMIT*-interface capacity
1 8	*SYSTEM LIMIT*-array rank
1 9	*SYSTEM LIMIT*-array size
1 10	*SYSTEM LIMIT*-array depth
1 11	*SYSTEM LIMIT*-prompt length
1 12	*SYSTEM LIMIT*-interface representation

Syntax Error		Value Error	
2 1	No array	3 1	Name with no value
2 2	Ill-formed line	3 2	Function with no
2 3	Name class		result
2 4	Invalid operation		
2 5	)CS not 0		

Implicit Argument Errors		Explicit Argument Errors	
4 1	□*PP ERROR*	5 1	*VALENCE ERROR*
4 2	□*IO ERROR*	5 2	*RANK ERROR*
4 3	□*CT ERROR*	5 3	*LENGTH ERROR*
4 4	□*FC ERROR*	5 4	*DOMAIN ERROR*
4 5	□*RL ERROR*	5 5	*INDEX ERROR*
4 7	□*PR ERROR*	5 6	*AXIS ERROR*

Access Control Vector   □*SVC*

1	x	x	x	Set by me require use by my partner.
x	1	x	x	Set by my partner require use by me.
x	x	1	x	Use by me requires set by my partner.
x	x	x	1	Use by my partner requires set by me.

Access States   □*SVS*

0	0	0	0	Not a shared variable.
0	0	1	1	Set by one processor and used by second processor (initial state).
1	0	1	0	Set by first processor, not yet used by second processor.
0	1	0	1	Set by second processor, not yet used by first processor.

Selected System Variable Values

$Z \leftarrow L \ \Box AT \ R$	**Attributes**
$L[1]$ Valence (0, 1, or 2)	$L[3]$ Exec. prop.(4 bits)
$L[2]$ Fix time (7 items)	$L[4]$ Object size (CDR)

$Z \leftarrow \Box NC \ R$	**Name Class**
$Z = {}^-1$ Invalid	$Z = 2$ Variable
$Z = 0$ Unused but valid	$Z = 3$ Function
$Z = 1$ Label	$Z = 4$ Operator

$\Box AI$	**Account Information**
$\Box AI[1]$ User Ident.	$\Box AI[3]$ Connect time
$\Box AI[2]$ Compute time	$\Box AI[4]$ Keying time

$\Box TS$	**Time Stamp**
$\Box TS[1]$ Year	$\Box TS[5]$ Minute
$\Box TS[2]$ Month	$\Box TS[6]$ Second
$\Box TS[3]$ Day	$\Box TS[7]$ Millisecond
$\Box TS[4]$ Hour	

$\Box FC$	**Format Control**
$\Box FC[1]$ Decimal Char.	$\Box FC[4]$ Fill for overflow
$\Box FC[2]$ Thous. indicator	$\Box FC[5]$ Print as blank
$\Box FC[3]$ Fill for blanks	$\Box FC[6]$ Neg. indicator

$\Box TC$	**Terminal Control Characters**
$\Box TC[1]$ Backspace	$\Box TC[3]$ Line feed
$\Box TC[2]$ New Line	

Selected System Function Arguments

$Z \leftarrow L \ \Box FX \ R$	**Fix (with execution properties)**
$L[1]$ No display	$L[3]$ Ignore attention
$L[2]$ No suspension	$L[4]$ Nonresource error is *DOMAIN ERROR*

$Z \leftarrow L \ \Box TF \ R$	**Transfer Form**
$L = 1$ (migration form)	$L = 2$ (extended form)

System Commands

```
)CLEAR  [size]
)CONTINUE  [HOLD]
)COPY  [libno] workspace [:[pswrd]] [names]
)CS[0|1|2|3]
)DROP  [libno] workspace  [:[pswrd]]
)EDITOR  [1, 2 or name]
)ERASE  names
)FNS  [first [last ]]
)HOST  command
)IN  file [names ]
)LIB  [libno] [:[pswrd ]] [first [last]]
)LOAD  [libno] workspace [:[pswrd]] [size]
)MCOPY  [libno] workspace [:[pswrd]] [names]
)MORE
)MSG[N]  userid message
)MSG  [OFF or ON]
)NMS  [first [ last]]
)OFF  [HOLD]
)OPR[N]  message
)OPS  [first [last ]]
)OUT  file [names ]
)PBS  [ON or OFF]
)PCOPY  [libno] workspace [:[pswrd]] [names]
)QUOTA
)RESET  [number]
)SAVE  [[libno] workspace[:[pswrd]]]
)SI
)SINL
)SIS
)SYMBOLS  [number]
)TIME
)VARS  [first [last ]]
)WSID  [[libno] workspace [:[pswrd]]]
```

Session Manager Commands

For a list of session manager commands, press **PF1**.

APL2 Editors

In the following paragraphs, *lines* is optional; if provided, it can be any of the following:

n1, n2, ..., n	Specifies lines *n1, n2, ..., n*
n1-n2	Specifies lines *n1* through *n2*
-n	Specifies lines 0 through *n*
n-	Specifies lines *n* through the last line

Command	Result
∇	Open/close definition
♥	Open/close locked definition
[→]	Abandon editing
[∆ *lines*]	Delete *lines*
[◻ *lines*]	Display *lines*
[*line* ◻ *position*]	Display and position (Editor 1 only)
[*line*] *text*	Replace or insert

Editor 2 only:	
[ ]	Continue statement
[▲] *expression*	Execute *expresssion*
[?] or PF1	Query PF keys
[ι] or PF2	Renumber lines
∇ or PF3	Close definition
[∇] or PF6	Save definition
[↑] or PF7	Scroll up
[↓] or PF8	Scroll down
[τ] or PF9	Scroll to cursor
[/string/ *lines*]	Locate text
[/string/ N *lines*]	Locate names
[/old/new/ ¨ *lines*]	Replace text
[/old/new/ ¨ N *lines*]	Replace names
[∨ *name lines*]	Put *lines* into *name*
[∧ *name lines*]	Get *lines* from *name*
[∧ *lines*]	Copy *lines*
[▲]	Comment

Invocation Options

In the following, *size* can be in bytes, kilobytes (K), megabytes (M), or percentage (%) form

```
AISIZE (size)    (TSO only)
       Maximum number of bytes stacked by AP 101.
APNAMES (name)
       AP's not automatically available
CASE (0|1|2)
       Defines alphabets allowed in APL names.
CODE (nnnn)
       Specifies the type of terminal you are using.
DATEFORM (ISO|US|EU)
       Specifies the format for time and date stamps.
DEBUG (nnn)
       Alters normal recovery actions of APL2.
DSOPEN (device)
       Specifies a value to be passed to GDDM.
EXCLUDE (device-token)
       AP's not to be loaded into your session.
FREESIZE (size)
       Minimum amount of free virtual storage
HILIGHT (INPUT OUTPUT ON|OFF)
       Specifies lines to be highlighted on the screen.
ID (nnnnnnn)
       Account number associated with the current session.
INPUT ('string')
       Specifies string(s) of data to be initially executed.
LOADLIB (dsname...)    (TSO only)
       Private LOADLIB for APL2 and AP's.
PROFILE (name)
       Name of session manager profile to be loaded.
QUIET
       No language processor output until prompted for input.
SHRSIZE (size)
       Amount of virtual storage reserved for the SVP.
SMAPL (ON|OFF|TRY)
       Specifies whether the session manager is to be used.
SVMAX (nn)
       Maximum number of concurrently shared variables.
SYSDEBUG (nnn)
       Special debugging settings for system programmers
TERMCODE (nnnnn)    (TSO only)
       Specifies the type of terminal you are using.
TRACE (nnn)
       Special debugging settings for system programmers.
WSSIZE (size)
       Storage (region) reserved for your active workspace
```

Auxiliary Processors

AP 100—Host System Command Processor

Shared Variables Req = 1

Initial Values:
```
CMS100←'[CMS|CP] [([370 EBCD])]'
TSO100←'[([370|EBCD])]'
```

AP 100 TSO Built-in Commands

`APL AI`: Accounting information
`APL ATTACH module[parameters]` Attach subtasks
`APL CODE [code]` Query and define terminal type
`APL DEBUG [code]` Query and set debug options
`APL DDI name` File information
`APL DSI name` Data set information
`APL LIB [n]` Workspace names
`APL LIBS` Library numbers
`APL QUIET` Suppress messages
`APL QUOTA` Returns information on limits
`APL USER` TSO information
`APL WSID`: Active workspace identification
`APL WSNAME [[libno] wsid]` DS name of ws
`EXEC clistname` Invokes the CLIST
`TSO commandname` Avoids conflicts between AP 100 built-in commands and TSO commands.

Return Codes:

-100	Command abended (CMS only)
0	No error
1	Command error
2	Command terminated by interrupt (TSO only)
3	Insufficient virtual storage (TSO only)
4	Insufficient shared storage (TSO only)
5	PDSI output is too large for the available buffer. (TSO only)
444	Invalid shared variable value

AP 101—Alternative-Input (Stack) Processor

Shared Variables: Req = 1

Initial Values:
```
SHR101←'([FIFO|LIFO])'
```

Fence Requests (TSO)

`SHR101←1`	Set or move the fence.
`SHR101←2`	Purge the stack up to the fence.
`SHR101←0`	Purge the entire stack.

Return Codes:

0	No error
1	Invalid initial value
12	Stack overflow (TSO)
444	Invalid shared variable value

AP 102—Main Storage Access Processor

Shared Variables: Req.=2.
AP 102 Service Requests:
$CTL102 \leftarrow 0$
    Obtain address of active workspace.
$CTL102 \leftarrow 1$ *address length*
    Obtain contents of *length* bytes from *address*
Protocol: CTL and DAT

Return Codes:
0	No error
2	No DAT variable offered
3	Invalid service request
4	Invalid number of parameters on service request
5	Invalid CTL variable value
6	Invalid address specified in CTL variable
7	Invalid length specified in CTL variable
8	Insufficient storage

AP 110—CMS File Processor

Shared Variables. Req.=1 or 2.
Initial Values:
$REC110 \leftarrow 'fileid[[FIX] [U|R|W] \{conversion\}'$
$CTL110 \leftarrow 'fileid \{CTL\}'$
The *conversion* can be: APL, BCD, BIT, $BYTE$, CDR, $COD1$, $DBCS$, $EBCD$, VAR, 370, or 192

Return Codes:
0	No error
1	File nonexistent or improper initial value
3	Permanent I/O error
4	First character of filemode invalid
5	Attempt to read too many records
6	Attempt to write too many records
7	Attempt to write past end of file
8	Attempt to read record of incorrect length
10	Attempt to create too many files
12	End-of-file read or read-only disk
13	Attempt to write on a full disk
14	Attempt to write on an unformatted disk
15	Attempt to write record of incorrect length to a fixed-length file
17	Record too large for output
19	Attempt to write in a full CMS file
22	Virtual storage exceeded
25	Insufficient storage
26	Item number invalid
27	Attempt to replace variable-length item with one of different length
443	Insufficient free storage
444	Invalid shared variable value
445	Insufficient shared memory for input

13

AP 111—QSAM File Processor

Shared Variables: Req = 1 or 2
Initial Value:
CMS: $REC111 \leftarrow$ ' ddname [conversion] '
TSO: $REC111 \leftarrow$ ' ddname   [L R W] [conversion] '
The conversion can be APL, BIT, BYTE, CDR,
COD1, DBCS, EBCD (or 192), TN (TSO only),
VAR, or 370 (or BCD).

Return Codes:
0	No error
1	Bad initial value or nonexistent file
12	End of file
15	Incorrect-length record on fixed-length output
17	Record too large for output
440	Error when opening file for output
441	Error when opening file for input
443	Insufficient space to process input
444	Invalid shared variable value
445	Insufficient shared memory for input

AP 120—Session Manager Command Processor

Shared Variables: Req = 2.
Initial Values: None.
Protocol: CTL and DAT
$CTL \leftarrow$ session manager command

Return Codes:
0 0	No error
1 11	Rank of control variable >1
1 13	Control variable contains noncharacter data
1 53	Required storage unavailable
1 56	Shared variable quota exceeded
1 60	Session manager not available
4 nnn	Error when command passed to session manager

AP 121—APL2 Data File Processor

Shared Variables: Req = 2
Initial Values: None
Protocol: CTL and DAT

AP 121 Service Requests

Service Request	Operation
' C fileid S '	Create a seq. file
' C fileid D '	Create a direct file
	Open for seq. write
' SWC fileid [length] '	APL2-only
' SW fileid [length] '	VS APL-compatible
' SR fileid '	Open for seq. read
' DR fileid '	Open for direct read
	Open for direct update
' DUC fileid '	APL2 only
' DU fileid '	VS APL-compatible
0 , recnum	Direct read
1 , recnum	Direct write
' ' or 1 0	Close a file
' D fileid '	Drop a file
' FS fileid size '	Change file size limit

14

AP 121 Return Codes:

0	No error
1	Record not found
2	File contains maximum records
3	Attempt to write an invalid object
4	Invalid object read
5	End of file
6	Conflict between file type and access
7	Record length error
9	Virtual disk or VSAM library full
11	Defined file size limit exceeded
12	Invalid service request
23	File must be opened for direct update
24	Improper library reference
25	File already exists
26	File does not exist
28	File in use by others
30	File protected from this operation
31	Maximum number of files open
32	No space for I/O buffer
33	Data variable not specified before write request
34	Incorrect or missing password
35	Hardware error or APL2 product failure
36	Named file is not an APL2 data file
37	File size limit reduced below previous limit
38	Invalid record length on sequential write request
39	Required record length not specified
42	Data variable not referenced
50	Invalid request for this subsystem

AP 123—VSAM File Processor

Shared Variables: Req. = 2.
Initial Values: None.
Protocol: CTL and DAT.

AP 123 Service Requests

Service Request	Operation
`'OR ddname[ :pswrd ]'`	Open file for read
`'OW ddname[ :pswrd ]'`	Open file for write
`'OU ddname[ :pswrd ]'`	Open file for update
`'OC ddname[ :pswrd ]'`	Open and clear reusable file
`'R'`	Sequential read
`'RU'`	Sequential read for update
`'R:key'`	Direct read
`'RU:key'`	Direct read for update
`'W'`	Write a record
`'W:key'`	Write new record (relative record)
`'E:key'`	Erase a record (key sequenced or relative)
`'PO[ :key]'`	Position record pointer
`'KF'`	Key feedback
`'C'`	Close file
`'T'`	Transfer data without translation (default)
`'T1'`	Translate to or from VS APL
`'T2'`	Translate to or from EBCDIC

AP 123 Return Codes:

1 12	Syntax error in request
1 13	VSAM file already open
1 15	Inappropriate VSAM open mode
1 16	Invalid service request
1 17	Key length error
1 18	VSAM MODCB error
1 19	VSAM SHOWCB error
1 20	Character vector required in data variable
1 21	Record length error in data variable
1 22	VSAM file not open
1 27	VSAM TESTCB error
1 32	Insufficient free storage for I/O buffers
1 33	No value assigned to data variable for write
1 42	Data variable not used yet
1 45	VSAM GENCB error
1 48	Invalid sequence of service requests

Selected VSAM Return Codes:

0 8	More than one record has the specified key.
4 116	File not properly closed after the last usage.
8 4	End of file, or specified key greater than highest key.
8 8	Duplicate key.
8 16	Record not found.
8 20	Record is being used by another user.
8 28	Data set is full.
8 32	Invalid relative byte address.
8 40	Insufficient virtual storage.
8 88	Sequential read without prior positioning.
8 96	Attempt to change key of record.
8 100	Attempt to change length of record on an entry sequenced or relative-record data set.
8 110	Attempt to open an empty file for read or update.
8 128	Attempt to open file not been properly allocated.
8 136	Insufficient virtual storage.
8 152	Password error.
8 168	Data set is in use by another job.
8 192	Invalid relative record number.

AP 126—GDDM Processor

Shared Variables: Req. = 2.
Initial Values: None.
Protocol: *CTL* and *DAT*.

AP 126 Service Request Codes

Code	Service Request
¯1	Query GDDM calls
¯2	Set error threshold
¯3	Set protection key
¯4	Set EBCDIC translation
¯5	Set default buffer size
¯6	Set AP 126 options
¯7	Query AP 126 options
¯8	Query subset of fields for modification
¯9	Query current hardcopy destination
¯10	Issue GDDM CHART Call

AP 126 Return Codes:

0 0	No error
1 7	Incorrect data variable length
1 11	$1 < ooCTL$
1 12	Syntax error in request
1 13	Character string specified in control variable
1 14	Invalid request code specified in control variable
1 21	$1 < ooDAT$
1 23	Numeric data specified in data variable
1 41	Data variable not specified
1 53	Required storage not available
1 56	Insufficient space in shared variable quota
1 60	GDDM not available
1 61	Invalid service request parameter
1 62	Invalid count, code, or length value on request
1 63	Hardcopy translate table not available
1 64	Reserved page id has been requested
1 65	Invalid hardcopy destination
1 66	Hardcopy destination not available
1 67	Hardcopy destination already open
ss nnn	GDDM return code nnn, severity ss
4 nnn	Warning; nnn is GDDM return code
8 nnn	Error; nnn is GDDM return code
12 nnn	Severe error; nnn is GDDM return code
16 nnn	Abend; nnn is GDDM return code

AP 127—SQL Processor

AP 127 allows you to use the Structured Query Language (SQL) on IBM DATA BASE 2 or SQL/DS. The SQL workspace is designed for use with AP 127.

AP 127 Operations $DAT \leftarrow 'CALL'$ name [values]
  Executes name and corresponding values.
$DAT \leftarrow 'CLOSE'$ name
  Closes cursor statement name.
$DAT \leftarrow 'COMMIT'$
  Makes all changes since last rollback or commit permanent.
$DAT \leftarrow 'EXEC'$ statement
  Immediately executes statement.
$DAT \leftarrow 'FETCH'$ name [options]
  Returns new result table data.
$DAT \leftarrow 'GETOPT'$
  Returns values of AP 127 options.
$DAT \leftarrow 'MSG'$ rcode
  Returns error message text.
$DAT \leftarrow 'NAMES'$
  Returns names of statements known to AP 127.
$DAT \leftarrow 'OPEN'$ name [values]
  Opens previously prepared cursor statement name.
$DAT \leftarrow 'PREP'$ name statement
  Prepares name for later execution.
$DAT \leftarrow 'PURGE'$ name
  Removes name from list of active AP 127 names.
$DAT \leftarrow 'ROLLBACK'$
  Discards all changes made since last rollback or commit.
$DAT \leftarrow 'SETOPT'$ options
  Sets values of AP 127 options.
$DAT \leftarrow 'STATE'$ name
  Yields current state of name.
$DAT \leftarrow 'STMT'$ name
  Yields statement name.
$DAT \leftarrow 'TRACE'$ n1 n2
  Yields an event trace of functions within AP 127.

Return Codes:

0 0 0 0 0	Normal return
0 1 0 *n n*	Warning message
1 0 0 *n n*	Error
1 1 0 *n n*	Transaction backout
0 0 1 0 0	Normal return, but may be more data.

AP 210—BDAM File Processor for TSO

Shared Variables: Req. = 2.

Initial Values:

$REC210 \leftarrow \text{'}ddname[[FMT]\ [U|R|W]\ [conversion]\text{'}$
$CTL210 \leftarrow \text{'}ddname\ (CTL\text{'}$

The *conversion* can be: 370 (or *BCD*), *BIT*, *BYTE*, *CDR*, *COD1*, *DBCS*, *EBCD* (or 192), *TN*, or *VAR*.

Return Codes:

-*nn*	File too small to contain records requested
0	No error
1	Invalid initial value
12	End of file
15	Wrong-length record on fixed-length output
17	Record too large
30	Record not found
440	Error in open for output
441	Error in open for input
443	Shared variable exceeds 32760 bytes
444	Invalid shared variable value
445	Insufficient shared storage for input data
901	I/O error in reading data